AMD

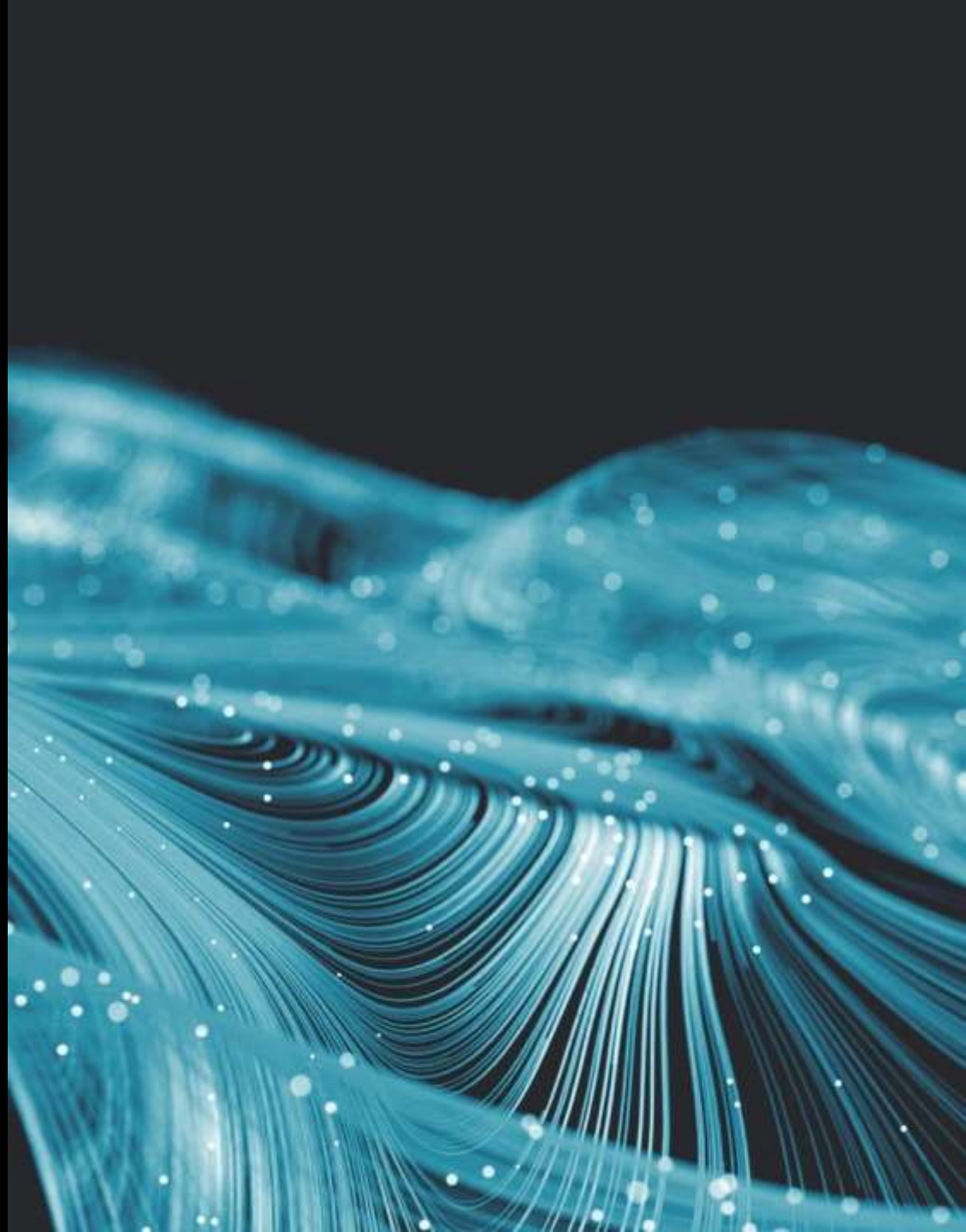
RUSTEC'2020

AMD ROCm™ and Radeon Instinct™: A Platform for High Performance Computing and Machine Intelligence

Timour Paltashev, D.Tech.Sci.
RTG MLSE, Senior Manager

October 2020

AMD PUBLIC

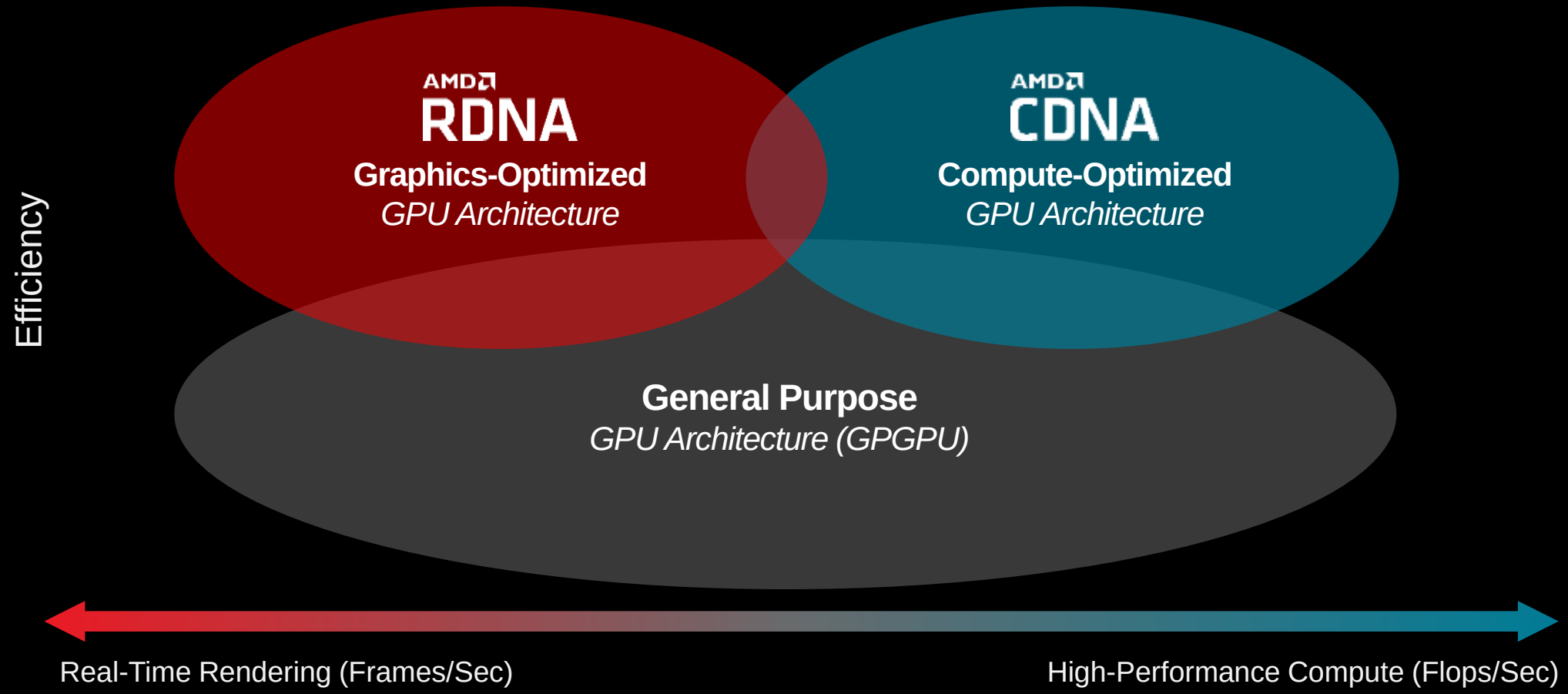


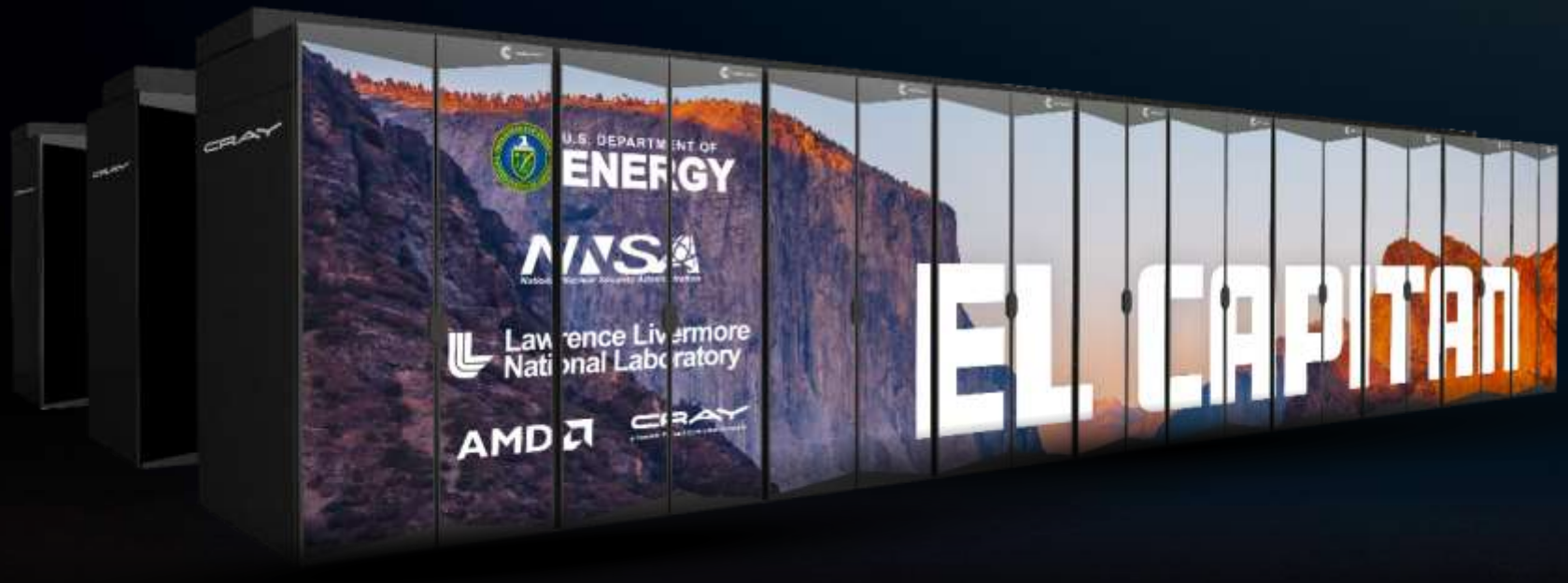
Cautionary Statement

This presentation contains forward-looking statements concerning Advanced Micro Devices, Inc. (AMD) including, but not limited to the features, functionality, availability, timing, expectations and expected benefits of AMD’s products, which are made pursuant to the Safe Harbor provisions of the Private Securities Litigation Reform Act of 1995. Forward-looking statements are commonly identified by words such as "would," "may," "expects," "believes," "plans," "intends," "projects" and other terms with similar meaning. Investors are cautioned that the forward-looking statements in this presentation are based on current beliefs, assumptions and expectations, speak only as of the date of this presentation and involve risks and uncertainties that could cause actual results to differ materially from current expectations. Such statements are subject to certain known and unknown risks and uncertainties, many of which are difficult to predict and generally beyond AMD's control, that could cause actual results and other future events to differ materially from those expressed in, or implied or projected by, the forward-looking information and statements. Investors are urged to review in detail the risks and uncertainties in AMD's Securities and Exchange Commission filings, including but not limited to AMD’s Quarterly Report on Form 10-Q for the quarter ended June 27, 2020.

Differentiated Strategy

Optimal Efficiency Through Domain-Specific Optimizations





AMD EPYC™ CPUs & Radeon Instinct™ GPUs Leading The Exascale Era

>2 ExaFLOPS Expected	Expected to be More Powerful than Today's 200 Fastest Supercomputers Combined	AMD Shipments in 2022

Source: https://www.top500.org/lists/top500/2020/06/

AMD CDNA Architecture

Compute DNA for the Data Center

Performance
Accelerate ML/HPC with Compute/Tensor OPS

Efficiency
Help Reduce TCO with High Perf-per-Watt

Features
Enhance Enterprise RAS, Security and Virtualization

Scalability
Scale Performance with AMD Infinity Architecture

What is ROCm™?

An Open Software Platform for GPU-accelerated Computing

AMD ROCm

Frameworks and Applications
TensorFlow, PyTorch, Caffe2

Libraries
MIOpen, roc* libraries

Programming models
HIP, C/C++, Python

Intermediate runtimes/compilers
LLVM based Clang, HCC

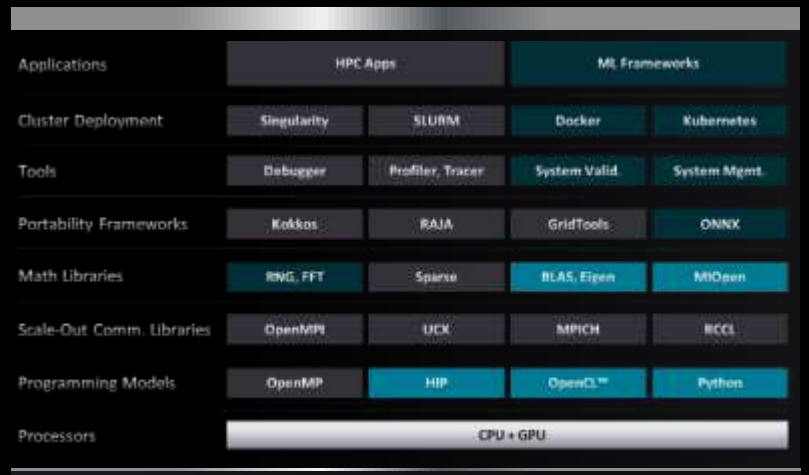
Runtimes
ROCm

Programmer and system tools
-debug
-profile

Data Center Software Evolution

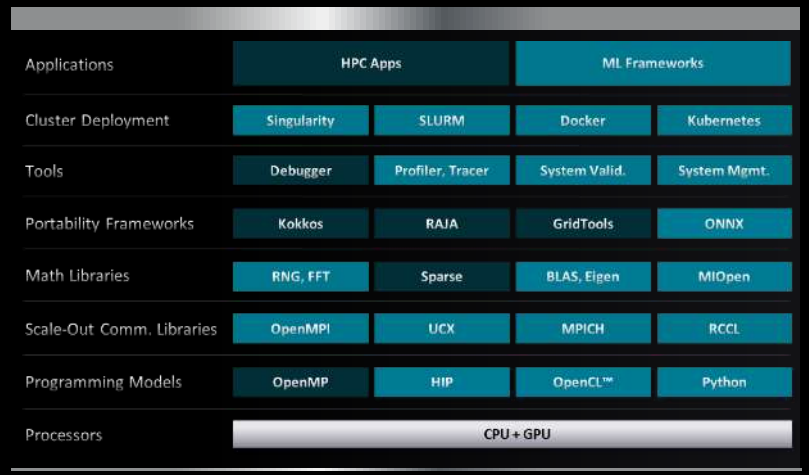
Steady Progress and Growing Ecosystem Support

AMD ROCm



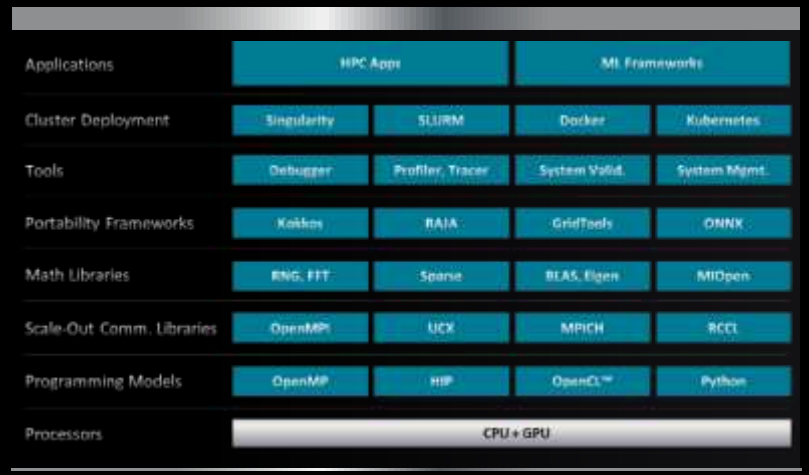
2018: AMD ROCm™ 2.0 Platform

Building the Foundation



2019: AMD ROCm™ 3.0 Platform

Focused on Machine Learning



2020 Plan: AMD ROCm™ 4.0 Platform

Complete Exascale Solution for ML/HPC

Future | Beta/Early | Production

Machine Intelligence

Natural Language Processing

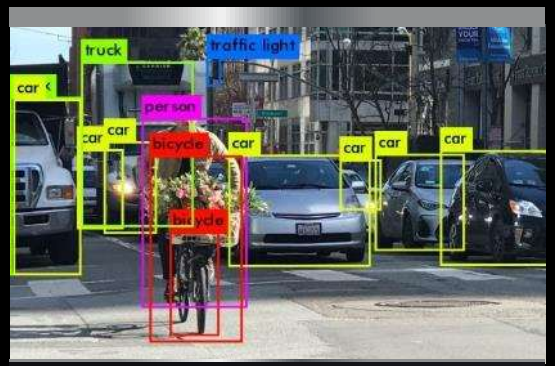


Image Recognition

Recommendation Engines

Industrial Automation

Revolutionizing Applications in Every Field

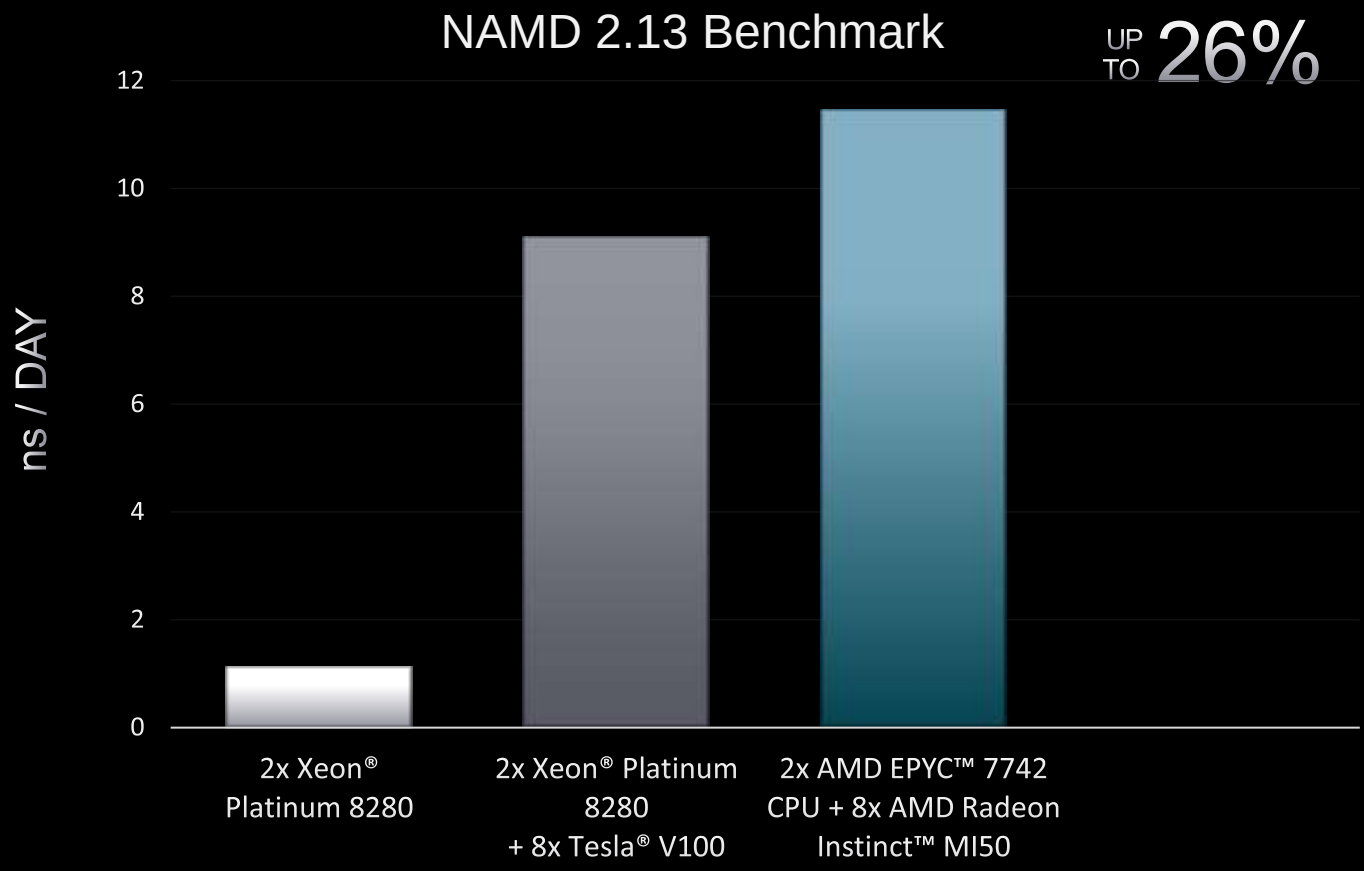
Exponentially Growing Demands for Performance

AMD Champions Open Source Solutions

AMD CPU + GPU + SW Advantages

Driving High-performance Computing Leadership

- Fully Integrated CPU and GPU Systems and Unified Tools
- Infinity Architecture for Bandwidth and Coherency
- Open Source Software Optimized for Performance



Testing Conducted by AMD performance lab as of 11-10-2019 using NAMD 2.13, STMV 1M Atom benchmark. Best-in-class based on industry-standard pin-based (LGA) X86 processors. Results may vary. (RIV-20)

HIP: Multi-Platform Capability for TCO Optimization

Easy to Deploy Porting Capability

CUDA-based application

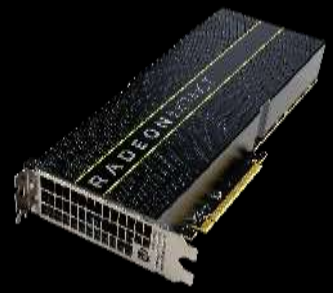
"HIPify"

Virtually Automatic Conversion

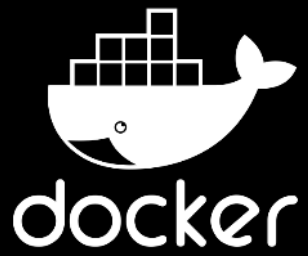
Portable HIP C++

Developer maintains HIP port

Resulting C++ code runs on NVIDIA or AMD GPUs



Fast-Growing ROCm™ Ecosystem



Containers

Sylabs Singularity

Data Center Workload Manager

Container Orchestration

Performance Profiling & System Tracer via PAPI

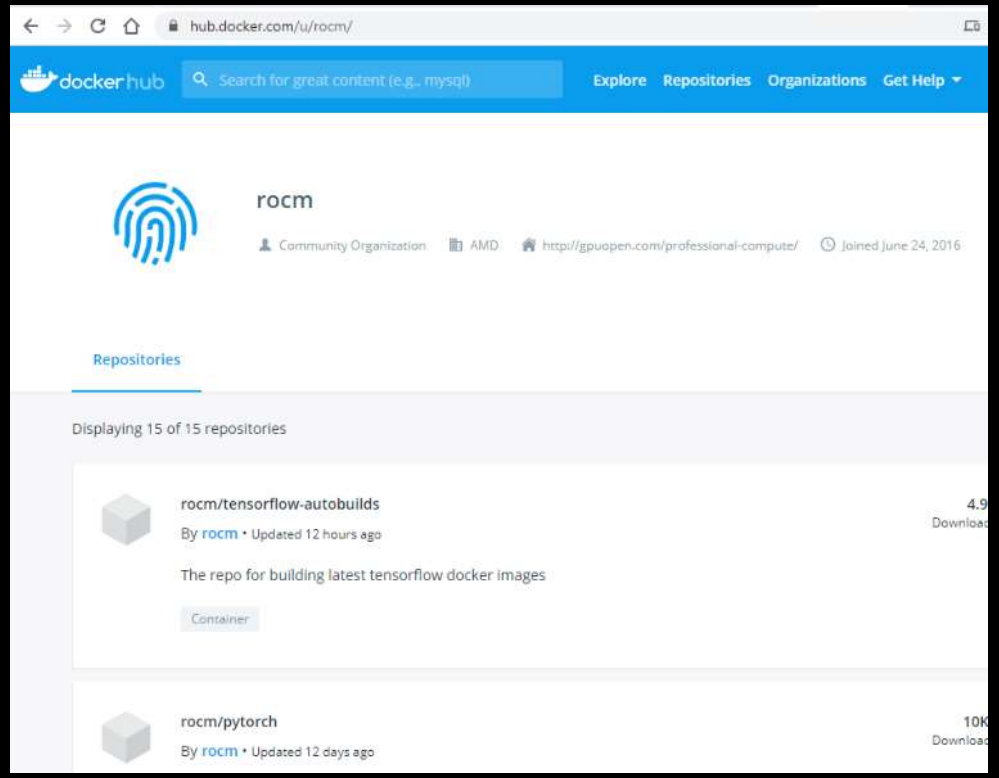
Eclipse C/C++ Development Tooling Based on ROC-GDB

Upstream ML Frameworks

Exascale Tools, Programming Models and Applications

Docker®

- Set permissions and add user to docker group
  - groups # identify the groups member
  - sudo usermod -a -G docker $LOGNAME
- ROCm™ Docker Hub
  - https://hub.docker.com/u/rocm/
- Run Docker Image
- docker run -it --network=host --device=/dev/kfd --device=/dev/dri --group-add video --cap-add=SYS_PTRACE --security-opt seccomp=unconfined -v /home/user:/home/user rocm/dev-ubuntu-18.04 bash
- Show running image
  - docker image ls
- Save container to your own image
  - Run docker commit on another terminal window
  - docker commit <container id> <my_docker_image>



Machine Learning Models

Deployable Today with Continuous Optimizations

Image Classification	Object Detection	Neural Machine Translation	Reinforcement Learning	Recommender Systems	Generative Models
• ResNet50/101 • ResNet152 • Inception3/4 • VGG16/19 • ShuffleNet • MobileNet • DenseNet • AlexNet • SqueezeNet • GoogleNet • ResNext101	• Faster-RCNN-ResNet50 • Mask-RCNN-ResNet50 • SSD-Resnet50	• GNMT: LSTMs • Translate: LSTMs • BERT: Transformer • GPT-2: Transformer	• Atari • Cart_Pole • VizDoom	• DLRM	• DCGAN • Fast Neural Style Transfer

AMD GPU Compilers:

C/C++

HIP (hip-clang)

- HIP (Heterogeneous Interface for Portability) is an interface that provides similar functionality to CUDA API
- Compiles HIP code and emits AMDGCN into binary
- hipcc -> hip-clang -> amdgcn
- Compiles to NVIDIA GPU with NVCC & its tool chain
- All the x86 pieces are dealt with in the same way

AOMP (AMD OpenMP Compiler)

- Compiles C/C++ code with OpenMP “target” pragmas
- Links with libomptarget to produce a binary that can offload work to the GPU

OpenCL™

- Khronos Industry Standard accelerator language

The GCN ISA is free and open!
https://developer.amd.com/resources/developer-guides-manuals/

AMD GPU Compilers: Fortran

OpenMP

- Support OpenMP 4.5+ target offload from FORTRAN with two open source options:
- F18 based on LLVM
- gfortran

HIP

- Offload kernels to GPU using Fortran 2003 C-binding
- hipfort project (in plan) to ease the wrap of GPU libraries

Frontier

- See Frontier spec sheet for what is expected to be supported: https://www.olcf.ornl.gov/wp-content/uploads/2019/05/frontier_specsheet.pdf

GPU Compilers Under Development: OpenACC

Mentor Graphics

- Has built GCC backend supporting the AMDGCN ISA

GCC

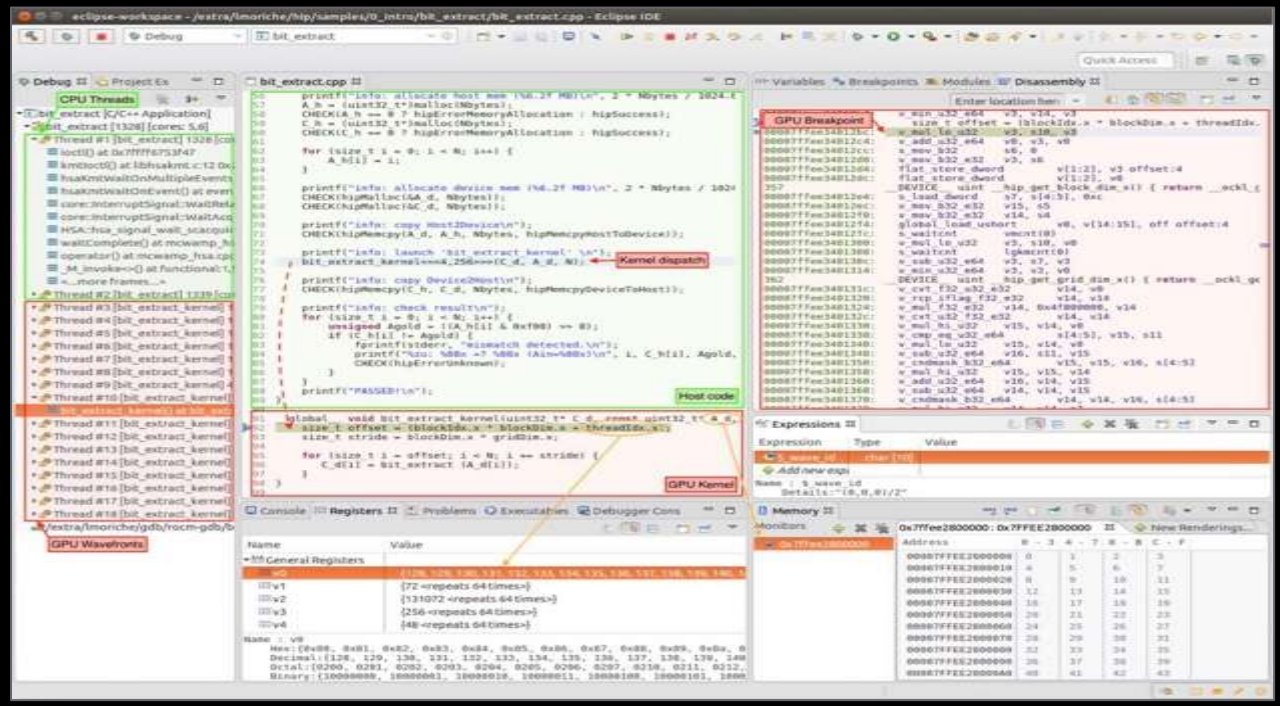
- OpenACC v2.6 is implemented in gcc and gfortran
- Mentor released updated compiler May 2020
- Optimizations and bug fixes ongoing – target Nov 2020 release

LLVM

- The Clacc project implements OpenACC in clang and can convert to OpenMP https://csmd.ornl.gov/project/clacc

Unified CPU & GPU Debugger

Easily Integrated with Industry Standard Tools



ROC-gdb
(gdb with GPU support)

RogueWave SOFTWARE
TotalView®

Linux Kernel
(ptrace)

ROC-dbgapi
(GPU Low Level Debug API Library)

GPU Kernel Driver

POLYTECHNIQUE MONTRÉAL

Released Q2-2020

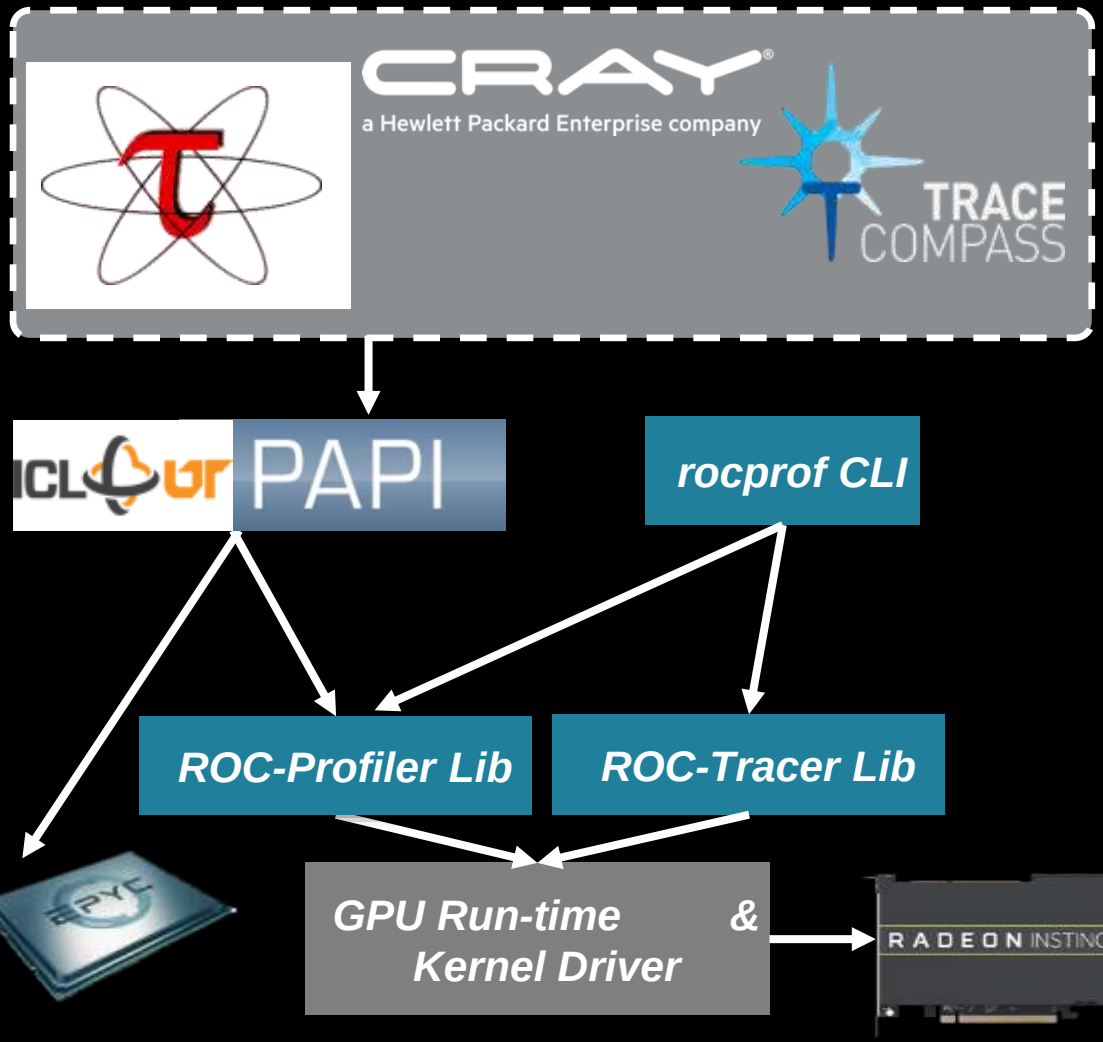
ROCgdb

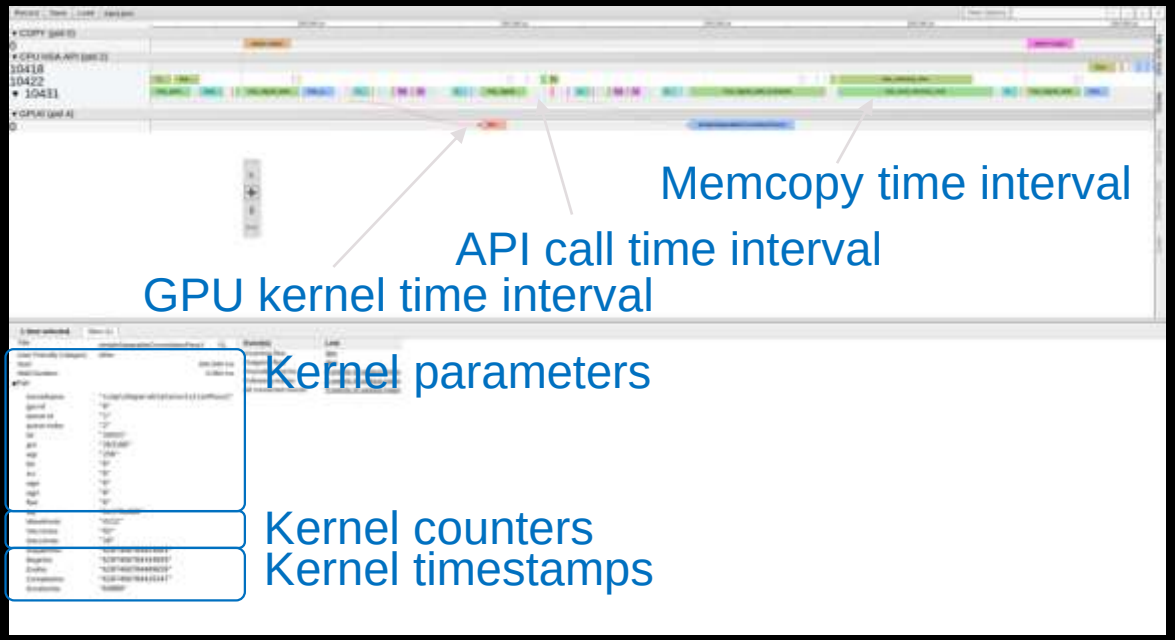
- ROCgdb is the ROCm source-level debugger for Linux
- ROCgdb is based on GDB, the GNU source-level debugger
  - https://github.com/ROCm-Developer-Tools/ROCgdb
- Compile executable using hipcc with “--ggdb"
- ROCgdb location:
  - `/opt/rocm/bin/rocgdb`
- To debug an executable
  - `rocgdb $EXE`
- To attach to a running process
  - `rocgdb -p <pid>`

```
(gdb) where
#0  0x000014825547ce57 in sched_yield () from /lib/x86_64-linux-gnu/libc.so.6
#1  0x00001482771954ad in amd::Event::awaitCompletion() () from /opt/rocm/hip/lib/libamdhip64.so.3
#2  0x00001482770ebc2d in ihipMemcpy(void*, void const*, unsigned long, hipMemcpyKind, amd::HostQueue&, bool) ()
   from /opt/rocm/hip/lib/libamdhip64.so.3
#3  0x00001482770ec127 in hipMemcpy () from /opt/rocm/hip/lib/libamdhip64.so.3
#4  0x0000000000409b08 in HPL_pdlange (GRID=<optimized out>, GRID@entry=0x7ffffcdf8770, NORM=<optimized out>,
    NORM@entry=HPL_NORM_1, M=<optimized out>, M@entry=45000, N=<optimized out>, N@entry=45000, NB=<optimized out>,
    NB@entry=384, A=<optimized out>, LDA=<optimized out>) at ../HPL_pdlange.cpp:302
#5  0x00000000004070ce in HPL_pdtest (TEST=TEST@entry=0x7ffffcdf8700, GRID=GRID@entry=0x7ffffcdf8770,
    ALGO=ALGO@entry=0x7ffffcdf8730, N=45000, NB=384) at ../HPL_pdtest.c:376
#6  0x0000000000402b33 in main (ARGC=<optimized out>, ARGV=<optimized out>) at ../HPL_pddriver.c:227
```

ROC-Profiler / Tracer

Easily Integrated with Industry Standard Tools





UNIVERSITY OF OREGON

ICL UT INNOVATIVE COMPUTING LABORATORY THE UNIVERSITY of TENNESSEE

POLYTECHNIQUE MONTRÉAL

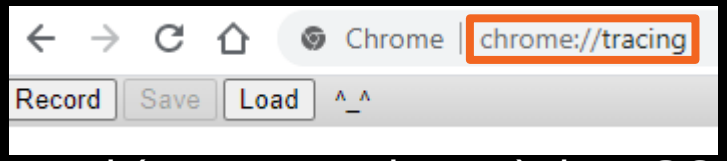
Released Q4-2019

rocprof

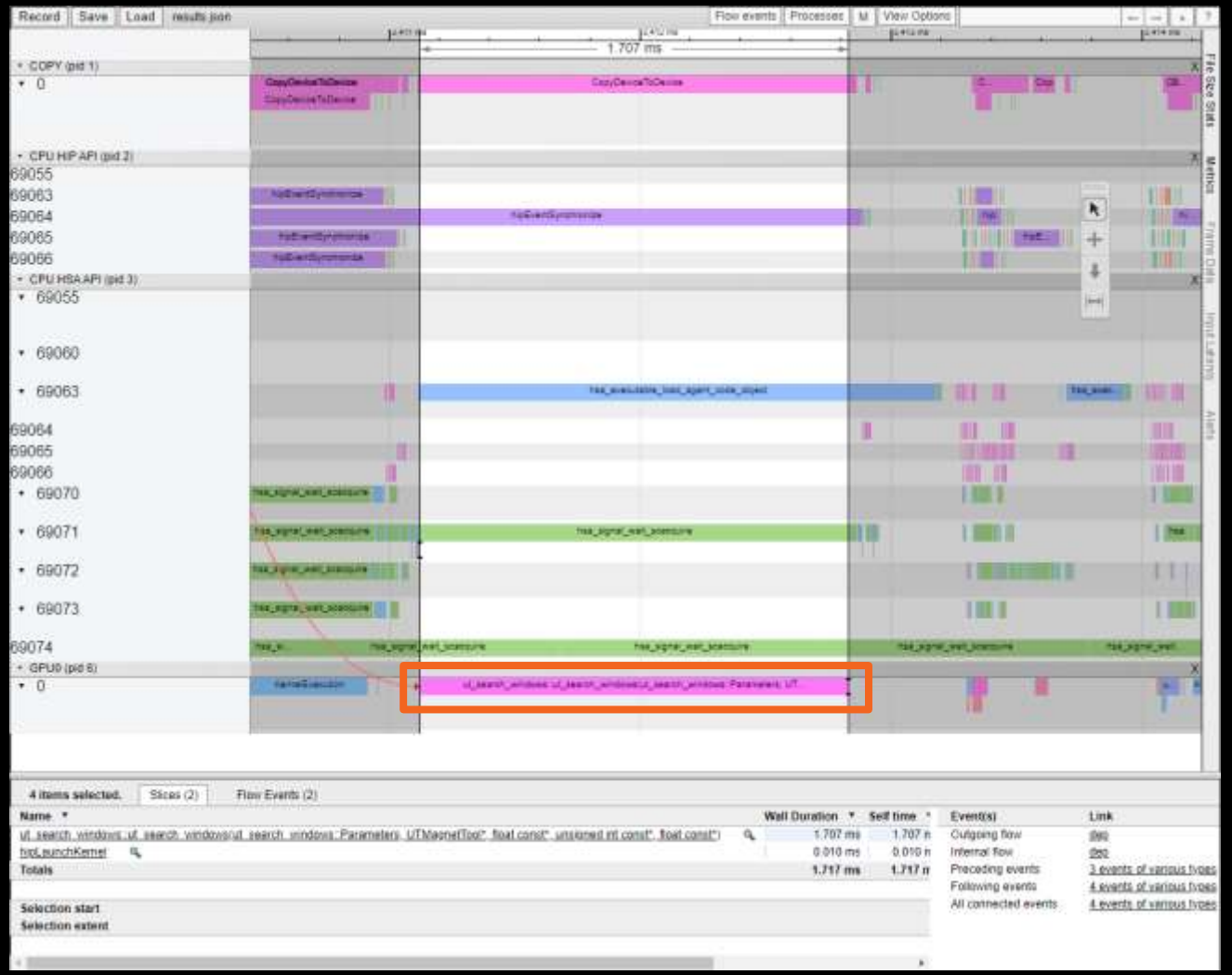
- rocprof is the AMD GPU profiler library
- It profiles with perf-counters and derived metrics
- To run rocprof to generate a kernel profile (text)
  - `rocprof --obj-tracking on --stats $EXE`
  - The default `results.stats.csv` will be generated
  - Comma-separated list of kernel activities

```
"Name","Calls","TotalDurationNs","AverageNs","Percentage"
KernelExecution,1614,473635087,293454,70.2867228678686
"lf_triplet_seeding::lf_triplet_seeding(lf_triplet_seeding::Parameters, LookingF
orward::Constants const*)",27,57000230,2111119,8.458746996112579
"velo_search_by_triplet::velo_search_by_triplet(velo_search_by_triplet::Parameter
rs, VeloGeometry const*)",20,27701080,1385054,4.110797925535989
"velo_calculate_phi_and_sort::velo_calculate_phi_and_sort(velo_calculate_phi_and
_sort::Parameters)",15,11600465,773364,1.721491272443271
```

- Run rocprof to generate a trace file
  - `rocprof --obj-tracking on --sys-trace $EXE`
  - Start Google Chrome
  - Type `chrome://tracing`



- Load (or Drag and Drop) the JSON file to view
- https://github.com/ROCm-Developer-Tools/rocprofiler/




```
--obj-tracking <on|off> - to turn on/off kernels code objects tracking [off]
  To support V3 code object

--stats - generating kernel execution stats, file <output name>.stats.csv

--roctx-trace - to enable rocTX application code annotation trace, "Markers and Ranges" JSON trace section.
--hip-trace - to trace HIP, generates API execution stats and JSON file chrome-tracing compatible
--hsa-trace - to trace HSA, generates API execution stats and JSON file chrome-tracing compatible
--sys-trace - to trace HIP/HSA APIs and GPU activity, generates stats and JSON trace chrome-tracing compatible
  '--hsa-trace' can be used in addition to select activity tracing from HSA (ROCr runtime) level
--kfd-trace - to trace KFD, generates KFD Thunk API execution stats and JSON file chrome-tracing compatible
  Generated files: <output name>.<domain>_stats.txt <output name>.json
```

Visit AMD.com/ROCm

Link to more training information:
https://community.amd.com/community/radeon-instinct-accelerators/blog/

Thank You!

Questions?

Disclaimers and Attributions

The information contained herein is for informational purposes only, and is subject to change without notice. Timelines, roadmaps, and/or product release dates shown in these slides are plans only and subject to change. “Polaris”, “Vega”, “Radeon Vega”, “Navi”, “Zen” and “Naples” are codenames for AMD architectures, and are not product names.

While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD’s products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale.

The information contained herein is for informational purposes only, and is subject to change without notice. While every precaution has been taken in the preparation of this document, it may contain technical inaccuracies, omissions and typographical errors, and AMD is under no obligation to update or otherwise correct this information. Advanced Micro Devices, Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document, and assumes no liability of any kind, including the implied warranties of noninfringement, merchantability or fitness for particular purposes, with respect to the operation or use of AMD hardware, software or other products described herein. No license, including implied or arising by estoppel, to any intellectual property rights is granted by this document. Terms and limitations applicable to the purchase or use of AMD’s products are as set forth in a signed agreement between the parties or in AMD's Standard Terms and Conditions of Sale. GD-18